International Institute for Applied Systems Analysis
IIASA www.iiasa.ac.at

Address: IIASA, Schlossplatz 1, A-2361 Laxenburg, Austria
Email: repository@iiasa.ac.at **Telephone:** +43 (0)2236 807 342

YSSP Report
Young Scientists Summer Program

Using the Future State Maximization paradigm to analyze the emergence of socially sub-optimal mobility behavior

Approved by

Supervisor: Dr.Nikita Strelkovskii
Program: Applied Systems Analysis
30.10.2020

This report represents the work completed by the author during the IIASA Young Scientists Summer Program (YSSP) with approval from the YSSP supervisor.

It was finished by ______, 31.10.2020 and has not been altered or revised since.

This research was funded by IIASA and its National Member Organizations in Africa, the Americas, Asia, and Europe.

ZVR 524808900

Table of Contents

Abstract....................iii
Acknowledgments....................iv
About the authors....................iv
Introduction....................1
Future State Maximization....................1
Traffic modeling as boundary object....................2
Method....................3
Environment....................4
An actual time-step....................4
A virtual time-step....................5
Agents....................5
Driving behavior....................6
Decision making....................10
Scenarios....................11
Results....................12
Sensitivity Analysis....................12
Horizon and number of walkers....................13
Horizon and the alpha parameter....................14
Patterns....................16
Discussion....................18
Influence of the three FSX parameters....................18
Patterns....................19
Bounded-rational intelligence....................19
Cooperation of FSX agents....................19
Computational demand....................20
Conclusions....................21
References....................23

Abstract

Agent-based models continue gaining popularity for simulations focusing on the microscopical description of social and other complex systems. Meanwhile, research on the method itself continues as novel findings in artificial intelligence excel the abilities to simulate individual decision making. Nevertheless, in between traditional rule-based ABMs and novel tools a lack for a transparent introduction of bounded rationality is imminent. Especially in microscopic traffic models a necessity for agents not entirely abiding by the traffic law can be discovered.

Recently, with Future State Maximization a promising description of intelligent behavior has been proposed. Agents making decisions according to this theory aim at maximizing the number of states they can reach in the future. This notion has so far been proven to enable the simulation of animals as well as humans. In both cases the agents exhibited behavior beneficial to the overall outcome, thus cooperating. If it is possible to also depict agents who do not necessarily cooperate, this method could prove to be a good fit in between rule-based and machine-learning-based ABMs.

We create a model of a road section populated with agents depicting cars. These entities are fitted with a future state maximizing decision making algorithm and face various scenarios. We then compare the results with common algorithms describing human driving behavior and assess the applicability of this new method to agent-based models in general.

Acknowledgments

First and foremost, I want to thank my YSSP supervisor Nikita Strelkovskii. He always had an open ear for ideas, he took an immense amount of time to rehearse presentations and discuss our findings. He taught me completely new perspectives and outright opened my eyes in one or the other case.

A big part in this regard is also taken by the entirety of the ASA group. Their efforts giving us YSSP participants the space and time to present our ideas really paid off. I was more than happy to be able to hear all of their comments and ideas. You really were an inspiration to me!

In general, the IIASA and especially the YSSP team have done great work to turn the YSSP into a truly transformative experience. COVID-19 left a tough year for most of us, but the effort of the YSSP team paid off in every regard. A big thank you to not canceling the event but turning all of the downsides of a pandemic into new possibilities.

None of this work would have been possible without the great support I received and still do by Prof. Manfred Füllsack. He believed in my abilities and laid the foundation of my success by guiding me along the path of academia. For the impulses and support in all our subject matter I can only be grateful to the whole working group informally known as the *AGF* (Arbeitsgruppe Füllsack).

Last but not least I also want to send thankful words to my family who supported me and gave me company during this socially distant summer times. I am happy to always have a home in the green to come back to.

About the authors

Simon Plakolb is a PhD candidate at the Systems Science group of the University of Graz. His primary research interests are the simulation of complex, social systems, using parallel computing, agent-based modeling, and artificial intelligence. As a boundary object for his investigations, he mostly uses traffic-related topics ranging from meso- to microscopically detailed models. His research touches the boundary in between computer science and social simulation, where implementation details on parallel hardware are as much of an issue as providing policy-relevant results. (Contact: simon.plakolb@uni-graz.at ; Homepage: www.behaviour.space)

Introduction

Agent-based modeling (ABM) is rising in popularity. More powerful computation technologies and an increase of microscopic simulations of complex, social systems brought forth ABMs ranging from topics as imminent as pandemics (Rockett et al., 2020) to more abstract use cases such as cardio-vascular systems (Bora et al., 2019).

While it is hard to clearly define the realm of agent-based modeling, a common understanding is, that individual, mobile entities independently acting upon a dynamic environment build the foundation of such a model (Grimm et al., 2005). Methodologically, these models are mostly implemented as rule-based models. The agents follow certain rules given by the modeler (Epstein, 1999). This sets such models into stark contrast to equation-based models which rely on differential equations to describe the model behavior (Van Dyke Parunak et al., 1998). The idea behind rule-based modeling is to provide an algorithmic description of the attributed behavior (Epstein, 1999; Jäger, 2019). In many cases of social simulation this attributed behavior is targeted at simulating intelligent decision making. Such intelligent agents are by some considered as the pinnacle of agent-based modeling (Bonabeau, 2002).

Compressing intelligent decisions into a set of rules however, has shown to cause some shortcomings (Jäger, 2019). Every behavior that is to be described requires its own specific rule (Epstein, 1999). Moreover, the set of rules is also inadvertently biased (Jäger, 2019). The intelligent decision of the agents is not so much simulated as it rather is defined as such. However, recently the idea has caught on to incorporate more advances algorithms in the agents (Manson et al., 2020). The past decade has brought about vast advances in the field of artificial intelligence (AI). It is reasonable to consider findings in this area as a possible solution for the lack of simulated intelligence in the agents.

Machine learning algorithms have gained in popularity in many research areas. As such they have also been incorporated in agent-based models to achieve artificial intelligence of agents (Jäger, 2019). Artificial neural networks prove to be capable of simulating behavior too complicated to be described in a reasonable set of rules. They can be fitted to existing data, yielding a matching behavior in the agents themselves (Jäger, 2019). However, such algorithms are essentially black boxes and prone to overfitting (Bilbao and Bilbao, 2017). Artificial neural networks can in principle exhibit the same result with vastly varying parameters. In some areas it may be favorable to rely on an algorithm which has a more direct input-output connection. One approach to this is to limit the number of input-output relations and provide clear meaning to them with Fuzzy Cognitive Maps (Kosko, 1986). In contrast to artificial neural networks and Fuzzy Cognitive Maps, both essentially systems of linearly connected rules, another solution could be *"one rule to rule them all"*.

Future State Maximization

The behavior of systems ranging from cosmological (Bousso et al., 2007) to biological and other domains (Martyushev and Seleznev, 2006) has recently been described using a single, fundamental, entropy-based formalism. Similar advances in AI research have incorporated this mechanism as an approach to approximate intelligence in known environments (Cerezo and Ballester, 2018; Charlesworth and Turner, 2019; Wissner-Gross and Freer, 2013).
The connection to the entropy-based advances in other fields has mostly been established by Wissner-Gross and Freer (2013). Their Causal Entropic Forcing (CEF) exhibited the first ever successful accomplishment of the string-pull test (Melis et al., 2006) by an artificial intelligence. This paper by Wissner-Gross and Freer (2013) sparked more in depth research into the matter. In effect,

CEF was used to simulate the formation of columns of people in a narrow passage way (Ebmeier, 2017) or the flocking of birds (Charlesworth and Turner, 2019). Hornischer et al. (2019) expanded their research to a more in depth analysis of interactions within CEF directed entities.

A similar approach to CEF was taken by Cerezo and Ballester (2018) who were more focused on the computational part in what they introduced as Fractal-AI. Their approach differs from CEF in that it does not primarily search for a maximum in entropy, but rather maximizes reward by so called walkers (Cerezo and Ballester, 2018). In physical terms, these walkers can be interpreted as particle representations of thought processes. Somewhat earlier Klyubin et al. (2005) pioneered the idea of CEF with their Empowerment prospect. Herein, the agent rather than maximizing the entropy of its state aims at maximizing its control over said state (Klyubin et al., 2005).

While there are some significant differences in between Causal Entropic Forcing, Fractal-AI and the Empowerment theorem, they still share a common principle. Charlesworth and Turner (2019) embrace an umbrella term to cover this general idea as Future State Maximization (FSX). We shall go on to use this term as an understanding of an agent which aims to enlarge its space of possible future states. A closer explanation of the principle can be found in the methods section below.

Traffic modeling as boundary object

Agent-based models are often considered where a microscopic description of human interaction is necessary. With the advent of autonomous vehicles it is paramount to confront their algorithms with realistic road usage scenarios. However, human driving behavior can only be depicted by including bounded rationality.

Traffic models range from macroscopic, top-down models to microscopic models of individual cars (Kotusevski and Hawick, 2009). The latter can be pictured as a form of agent-based models (Hidas, 2002). Commonly the movement of the cars is simulated using algorithms based on the car-following model by Gipps (1981) (Ciuffo et al., 2012). These models generally share the rule-based approach with other common agent-based models. Thus, agents mostly adhere to traffic laws, which implies problems introducing bounded rationality (Jäger, 2019).

While bounded rationality shows effects on emissions, these shortcomings do not infringe the overall outcome vastly on the larger scale of most traffic models (Tang et al., 2015). However, for certain applications it is paramount to investigate such behavior i.e. shared spaces without clear lanes to follow, or autonomous car algorithms taking into account human driving behavior.

Microscopic traffic models of road sections, therefore, can act as a good boundary object to assess the capabilities of Future State Maximization to remedy the outlined shortcomings.

However, a striking resemblance in the FSX models proposed to date is the emergence of cooperative behavior. Be it by column formation (Ebmeier, 2017), providing aerodynamic benefits (Charlesworth and Turner, 2019) or successful attempts at a test aimed to examine cooperation of animals (Melis et al., 2006; Wissner-Gross and Freer, 2013). Realistic agent-based models can only be deployed if these effects can also break down and bounded rationality can lead to detrimental effects (Jäger, 2019). In the case of our microscopic traffic model, cooperation should lead to the agents reaching their destination as quick as possible. In the event of an obstacle blocking one lane, cooperative behavior is key for quick and effective merging onto the other, populated lane (Bae et al., 2020). Thus, our agents need to take advantage of others cooperative behavior (Bae et al., 2020) while also acting cooperative towards others. In effect, a short travel time without collisions is the collective objective.

Method

As Future State Maximization embodies a common principle found in multiple applications, no clear definition of an algorithm exists. No comparison of the various implementations published to date exists as of the writing of this report. However, all of the published examples exhibit somewhat intelligently behaving agents.

Cerezo and Ballester (2018) present an intuitive and comprehensive description of an algorithm for their variant. This was utilized as a guide line for the methodological part of the work at hand. Formalisms found in papers stemming from the Causal Entropic Forces line of FSX-research focus on calculation of state entropy (Hornischer et al., 2019). In comparison, the algorithm of Fractal-AI is more in line with the perception of "maximizing future states" (Cerezo and Ballester, 2018; Charlesworth and Turner, 2019). It combines the notion of walkers as entities sent into the future state space with a mechanism for the (re-)distribution of these. Thus, in contrast to causal entropic forcing (Wissner-Gross and Freer, 2013) it features the ability to introduce a utility function (Cerezo and Ballester, 2018). What had previously been a method to sample the future state space of the agent (Hornischer et al., 2019), also referred to as its causal cone (see Figure 1), is put into focus by (Cerezo and Ballester, 2018).

Walkers are created whenever the agent faces a decision. They initially reside at the very state of the agent they are spawned from. Consequently, according to Fractal-AI (Cerezo and Ballester, 2018), each walker draws and performs one random action a_0 from the agent's action space. The action space can be either discrete or continuous. Each initial action is stored with the identity of the walker performing it as a crucial part of the walker's integrity (Cerezo and Ballester, 2018). Note that any walker now finds itself at a new state $s_{t+1} = f(s_t, a_0)$. Note that f is required to be injective.

This state can now be evaluated as featuring the walker either alive or consumed via an evaluation function g. As such, g maps the state space to the binary result of *dead* or *alive* and is thus also required to be injective. Dead walkers are not abandoned but re-initiated. For each walker that seizes to be alive a random living walker is duplicated, yielding two walkers with identical initial choice and walker state (Cerezo and Ballester, 2018). Henceforth, each walker draws another random action a_1 and the process repeats. Similar to CEF, the iterative process ends at the so-called horizon $\tau \in \mathbb{N}_0$ where walkers have ultimately performed action a_τ and reached their final walker states (Cerezo and Ballester, 2018; Hornischer et al., 2019; Wissner-Gross and Freer, 2013).

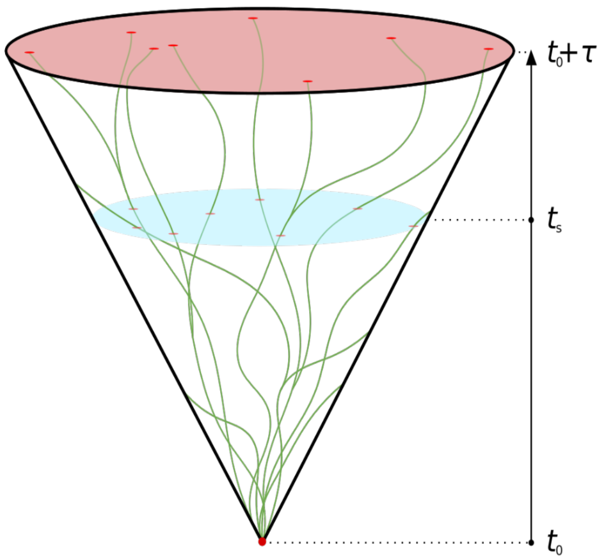

Figure 1: The causal cone of an agent and walkers scanning it up to the agent's horizon τ. The agent's state is depicted as red dot at the tip of the cone, whereas the walkers are shown as green strings. At t_s a causal slice can be observed.

This can be followed through in Figure 1 where an Agent at t_0 spawns 8 walkers. Note that once walkers (depicted as green strings) hit the edge of the cone and thus cease to exist, a living other walker duplicates.

In effect, even though walkers collide with the boundary in the process, the algorithm ensures that the amount of walkers stays constant.

As a result, the distribution of the walkers approximates the distribution of future states of the agent which can be reached within the horizon and are mapped into the survivable realm by g. The consequent action of the agent spawning these walkers is then the choice with the most walkers surviving in the discrete action space or the average of all surviving walkers in the continuous case (Cerezo and Ballester, 2018).

A major advantage of the Fractal-AI algorithm is the possibility of incorporating a utility function (Cerezo and Ballester, 2018). This is accomplished by skewing the walker distribution such that it aligns with the distribution of reward as given by the utility function (Cerezo and Ballester, 2018). Obviously, the effective reward in any future state can only be known where a walker is placed. Hence, somewhat in contrast to causal entropic forcing, Fractal-AI relies on a sufficient number of walkers to arrive at an appropriate scanning density. This shall become evident in the sensitivity analysis found in the results section.

Environment

A notion of state relies on an environment it resides in. The environment utilized to inhabit the car-like agents is based on an infinite, two-dimensional, Cartesian coordinate system. Within these coordinates exist areas which can be understood as accessible. Any area outside these renders a walker interfering with it dead. Other agents are also considered as inaccessible areas. Hence, the agents are motivated to stay within the bounds of the accessible areas, yet clear of other agents.

As Cazzolla Gatti et al. (2020) argue, any action taken, and thus any decision, sparks an infinite number of future possibilities. Therefore, it is paramount to distinguish two different ways for the environment to evolve:

- Actual time-steps (for agents)
- Virtual time-steps (for walkers)

The prior resembles the intuitive case in which every agent executes its decision at time t and consequently arrives at a new state in $t+1$ impacted by this decision. However, prior to this execution, the decision must be found. At this stage the actual development of the environment is unknown. The virtual time-step consequently reveals an approximation of future time-steps. Therefore, it is possible to incorporate an agent's beliefs about future development. Within this report only agents with equal beliefs are considered. This common expectation is that fellow agents follow their trajectory linearly. However, it would prove trivial to introduce e.g. the belief that others will slow down if oneself does not for particular agents.

An actual time-step

The movements of the agents are processed sequentially. Every agent is subsequently called to decide whether to brake or accelerate and where to steer. These decisions are then executed such that all agents arrive at an agent state s_{t+1}.

Every single agent thus spawns a number of walkers which in turn perform virtual time-steps through the estimated future state space. In contrast to the actual time-step which re-positions agents these virtual time-steps do not alter the environment of the agents. Nonetheless, the dynamics of the environment including movement of other agents are estimated within a virtual time-step.

A virtual time-step

In order to find a dynamic view into the future state space, the trajectories of all agents are pursued. Therefore, we arrive at a linear development of the difference of the two prior states. Thus, virtual states vs_n can be found through actual states as_t and as_{t-1}. For a well defined sum and difference of states the initial virtual state vs_0 could be denoted as:

$$\mathrm{vs}_0 = \mathrm{as}_t + \Delta \mathrm{as}_t \qquad \text{(Equation 1)}$$

Where $\Delta \mathrm{as}_t = \mathrm{as}_t - \mathrm{as}_{t-1}$. All further virtual states are then found by continuing these trajectories, as in:

$$\mathrm{vs}_n = \mathrm{vs}_{n-1} + \Delta \mathrm{as}_t \quad | \quad n > 1 \qquad \text{(Equation 2)}$$

These virtual time-steps construct the environment scanned by the walker entities. The future states a walker experiences may differ from the actual state space the agent ultimately encounters. Hence, a virtual time-step can be regarded as the expectations the agent has about the future development of its surroundings.

Agents

The agents, stylized cars, are rectangular with a length l_a and a width w_a. Their movement can be forwards or backwards with the limited speeds v_{fwd} and v_{bwd}. A minimum absolute speed v_{min} is required to turn in any direction. However, the vehicle turning angle is limited by δ_t. The pivot point for turning is set in the center back of the agent as depicted in Figure 2.
Acceleration and deceleration are idealized as being linear and limited by a_{fwd} and a_{bwd}. The movement of the walkers is restricted by the same parameters.

Table 1: Parameters for the car-agent properties.

Parameter	Name	Value	Unit	Original value (Gipps, 1981)
Length	l_a	4.0	m	~ 6.5
Width	w_a	1.8	m	-
Max. Speed forwards	v_{fwd}	24.0	m/s	~ 20.0
Max. Speed backwards	v_{bwd}	3.0	m/s	-
Min. turning speed	v_{min}	1.0	m/s	-
Max. turning angle	δ_t	0.28	rad	-
Max. acceleration	a_{fwd}	3.0	m/s^2	~ 1.7
Max. deceleration	a_{bwd}	6.0	m/s^2	2 * a_{fwd}

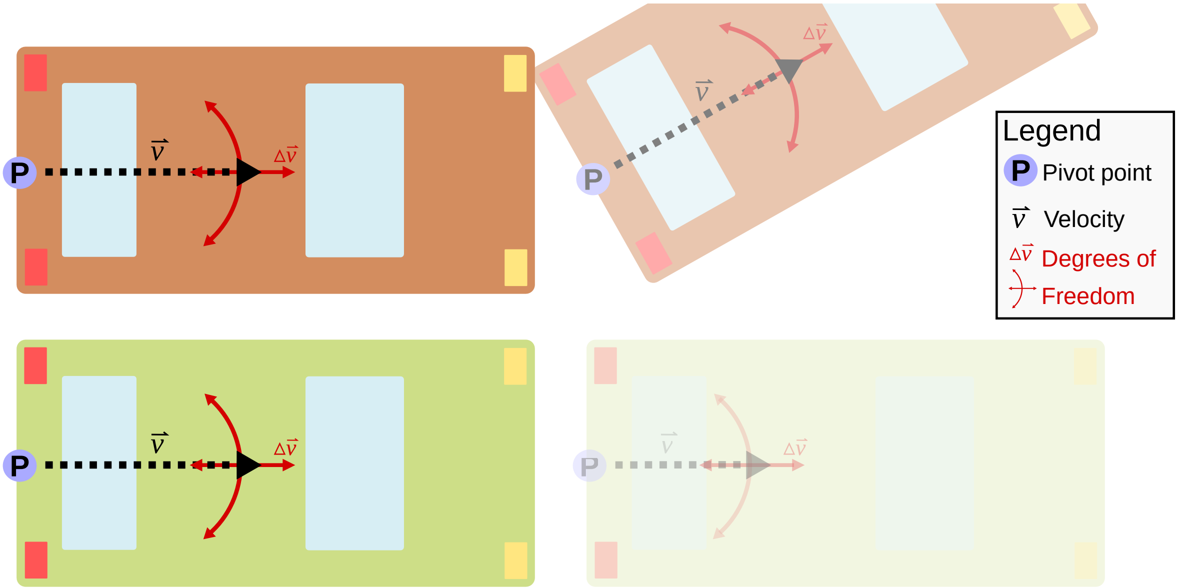

Figure 2: Two agents and their velocity before and after they have come to a decision. The orange agent decides to turn left, while the green agent decides to decelerate.

All of these parameters of movement were set to closely resemble the parameters of Gipps (1981) while also taking into account an average modern car. Their values are listed in Table 1.

As the agents in the model are future state maximizing agents, three additional parameters for the decision process have to be considered: The horizon τ, the number of walkers N and the urgency to reach the objective α. While the parameters of movement will remain constant, these three parameters remain subject of investigation throughout this report.

Driving behavior

The driving behavior of the agents emerges from the set limits and their decision process. The decision is composed of two objectives: Survival as given by g and a reward function. The achievement of both objectives follows a similar approach. In contrast to Cerezo and Ballester (2018), we decided to refrain from a single reward function encoding survival and utility functionality. This choice was taken in order to have a simpler mechanism to balance and assess the influence of each single objective.

Survival is the primary objective. The state variable encoding the survival of a walker is binary: It equals 1 if the walker is alive and 0 if it was consumed. After all walkers have taken a step in to the future state space, those walkers intersecting or going beyond the boundary of the accessible area are marked as consumed.
Each consumed walker is then replaced by a duplicate of a random surviving walker as laid out in Algorithm 1.

```
// Start: Collect all of the N walkers which stayed alive
walkers_new := Set()
FOR i := 0 TO N DO BEGIN
    IF walkers(i).alive
    // Add alive walkers to "walkers_new"
    THEN walkers_new.add(walkers(i))
    END
END
// Determine how many walkers were caught dead
missing := walkers.length – walkers_new.length
// Fill up the now missing walkers with new walkers
FOR j := 0 TO missing DO BEGIN
    // Create a new walker entity
    clone  := Walker()
    // Choose a random walker out of the surviving ones
    random_walker := random_choice(walkers_new)
    // Copy the state and the initial decision into the new one
    clone.state = random_walker.state
    clone.init_decision = random_walker.init_decision

    walkers_new.add(clone)
END
```

Algorithm 1: Ensuring a stable number of walkers by cloning living walkers upon the death of a walker.

Additionally, a reward is introduced to motivate the agents to move. This reward function r is defined as $r(d) = \frac{1}{d}$ where d is the distance of the walker's position x_i to the position of the agent's goal $|x_g|$:

$$d = |x_g - x_i| \qquad \text{(Equation 3)}$$

To redistribute the walkers according to the reward density, Cerezo and Ballester (2018) then introduce a quantity called "Virtual Reward" VR_i. This normalized measure of individual reward enables comparison among walkers (Cerezo and Ballester, 2018).

$$\mathrm{VR}_i = \frac{R_i}{D_i} \qquad \text{(Equation 4)}$$

where D_i is the density of walkers in the vicinity of the state of walker i and R_i is the reward. In order to align the walker density with the reward density the goal is therefore to balance the virtual reward among all walkers (Cerezo and Ballester, 2018). This is achieved via a randomized redistribution process.

Firstly, Cerezo and Ballester (2018) approximate D_i in order to spare computational demand by finding the distance to a random other walker:

$$D_i = Dist(w_i, w_j) \text{ with } i \neq j,\ j \text{ randomly chosen } \in [1, N] \qquad \text{(Equation 5)}$$

Then, normalization functions $Norm$ and $Scale$ are introduced:

$$Norm(x, \sigma, \mu) = \frac{x - \mu}{\sigma} \qquad \text{(Equation 6)}$$

$$Scale(x) = \begin{cases} 1 + \ln(x) & \text{for } x > 1.0 \\ e^x & \text{for } x \leq 1.0 \end{cases} \qquad \text{(Equation 7)}$$

where μ is intended to denote the mean of the distribution x is an element of and σ its standard deviation respectively.
Finally, the virtual reward can now be calculated as:

$$\mathrm{VR}_i = Scale(Norm(R_i))^\alpha * Scale(Norm(D_i)) \qquad \text{(Equation 8)}$$

Equation 8 introduces an additional parameter $\alpha \in [0, 2]$. This parameter is intended to be used to balance exploitation versus exploration (Cerezo and Ballester, 2018). The idea is that a high α increases the weight of the reward. Hence, agents will increasingly strive for reward as α increases. With $\alpha \to 0$ the entropy of the walkers is emphasized. Hence, such agents explore their environment disregarding the areas with higher pay-off. However, with a low α the algorithm given by Cerezo and Ballester (2018) also loses information about the evaluation function g. Thus, the agent performs undirected exploration and is prone to run into states which are not survivable (Ishii et al., 2002). With our modification a more efficient, directed exploration (Ishii et al., 2002) can be performed as information about survival persists.

To accommodate our second objective, these virtual reward values can now be used to distribute the walkers according to reward density. This mechanism is similar to Algorithm 1. However, in contrast to the survival objective, a probability p of a walker being substituted is introduced:

$$p_i = \begin{cases} 0 & \text{for } \mathrm{VR}_i > \mathrm{VR}_j \\ 1 & \text{for } \mathrm{VR}_i = \ 0 \\ \frac{\mathrm{VR}_j - \mathrm{VR}_i}{\mathrm{VR}_i} & \text{for } \mathrm{VR}_i \leq \mathrm{VR}_j \end{cases} \qquad \text{(Equation 9)}$$

This probability again relies on a randomly drawn walker j where $j \neq i$. A complete lack of virtual reward immediately schedules the walker to be substituted, whereas a superior virtual reward ensures its survival. As also outlined in Algorithm 2, in the case of an inferior virtual reward to the randomly drawn competing walker, a normalized difference in the virtual reward establishes the probability of the walker i to be substituted.

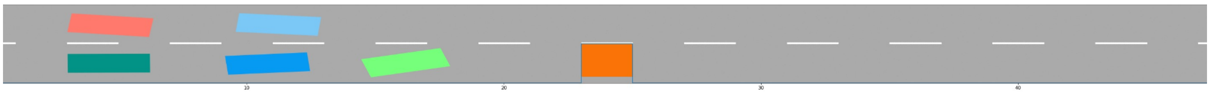

Figure 3: Five agents approaching an obstacle blocking the right lane in the primary scenario.

```
// Initialization: Create a set for all of the N walkers to continue scanning
walkers_new := Set()
// Traverse over all N walkers
FOR i := 0 TO N DO BEGIN
    // Draw a random walker to compare against
    random_walker := random_choice(walkers)

    // Get the virtual reward of both the walker at i and the sampled one
    VR_i := walkers(i).virtual_reward
    VR_r := random_walker.virtual_reward

    // Compare the virtual rewards
    IF VR_i > VR_r
    THEN
        // If the walker at i has superior reward it will continue
        p := 0
    ELSE
        IF VR_i == 0
        THEN
            // If the walker at i has no VR it will be replaced
            p := 1
        ELSE
            // In all other cases the probability of replacement
            // is determined as normalized difference
            p := (VR_r – VR_i) / VR_i
        END
    END
    // REPLACEMENT
    // Draw a random number, evenly distributed in between 0 and 1
    r := rand()
    // Compare p to r
    IF p < r
    THEN
        // Add the walker at i to the new set of walkers
        walkers_new.add(walkers(i))
    ELSE
        // Create a new walker entity
        clone  := Walker()
        // Copy the state and the initial decision into the new one
        clone.state = random_walker.state
        clone.init_decision = random_walker.init_decision
        // Add the clone to the new set of walkers
        walkers_new.add(clone)
    END
END
```

Algorithm 2: Walker redistribution taking into account a reward function.

Decision making

With the walker redistribution functions defined, the formulation of an agent's thought process is straight forward. As our model operates in continuous state-space we will focus on the description of this specific case.

```
// Start: Initialize N walkers
walkers := Set()
FOR i := 0 TO N DO BEGIN
    // Initialize a walker entity
    walker := Walker()
    // Copy the parent agent's state to the new walker
    walker.state := self.state
    // Sample a random action and store it as initial action
    walker.init_action := DecisionSpace.sample()
    // Perform the initial action
    walker.perform(walker.init_action)
    // The walker is now at a new state at t+1
    // and added to the set of walkers
    walkers.add(walker)
END
// Thought process: Virtual time-steps until horizon τ
FOR h := 1 TO τ DO BEGIN
    // Perform Algorithm 1
    // Dead walkers are now clones of living walkers,
    // including their initial action
    walkers := surviving_walkers(walkers)
    // Perform Algorithm 2
    // Poorly performing walkers are now clones of better performing ones,
    // including their initial action
    walkers := rewarded_walkers(walkers)
    FOR i := 0 TO N DO BEGIN
        // Sample another random action for each walker
        // Note that this random action is not stored
        walker(i).perform(DecisionSpace.sample())
    END
END
// Decision: The final set of walkers dictates the agent's decision.
//           In the continuous case this is the mean of all initial actions.
decision := mean(walkers.init_actions)
```

Algorithm 3: Ensuring a stable number of walkers by cloning living walkers upon the death of a walker.

The agent spawns N walkers which initially are all situated at the agent's position in state space. They then each perform a random action which is stored as the initial action. Then they are re-distributed as given in Algorithm 1 according to their survival. Next the new walker entities are re-distributed taking into account their virtual reward. Thus, aligning the walker distribution with the reward distribution (Cerezo and Ballester, 2018) given by the utility function as outlined in Algorithm 2. Consequently, the walkers perform another random action.
This process is repeated until the agent's horizon τ is reached. Since in Algorithm 1 and Algorithm 2 upon the cloning of a walker its initial action is also duplicated, the final distribution of initial decisions

is skewed towards walkers which (a) survived and (b) achieved a higher reward. Ultimately, the decision of the agent is the mean of all initial decisions of the walkers contained in the final set. A detailed description of this decision mechanism is given in Algorithm 3.

Scenarios

Three different scenarios where developed to assess the performance of the FSX principle. The basic scenario is an open road with a width of 6 meters. This width approximately accounts for a two lane road. A variant of this scenario with a single lane road exists too. The road has no formal end but loops after 200 meters.

The primary scenario depicted in Figure 3 is built on a similar road with a single object blocking one lane. While a secondary scenario introduces a second obstacle on the opposite lane in some distance to the first obstacle. In both cases the road does not loop but end. The goal of the agents is to go beyond the obstacles as quick as possible. Since the objective is to cooperate, only the time of the last agent to pass the obstacle course is considered. Simultaneously, the agents also want to avoid collisions due to Algorithm 1. The overarching objective is thus to collectively pass the obstacles in short time with as little collisions as possible. A further explanation of how these values interact can be found in the results section. The scenarios are listed in Table 2.

Table 2: List of scenarios and the properties of their environments.

Scenario	Number of lanes	Obstacles	Road length (m)
Primary scenario	2 á 3 m	1; blocking the right lane at 23 m	200
Secondary scenario	2 á 3 m	2; one on the right lane at 23 m, the other at 48 m blocking the left lane	200
Basic scenario	2 á 3 m	-	infinite
Single lane	1 á 3m	-	infinite

The starting coordinates are the only distinct feature in the setup of the agents. These coordinates mark the pivot point of the agent as outlined in section “Agents” and depicted in Figure 2. The initial placement puts 3 meters in between the agents in their driving direction. They all face towards the same direction which is also the direction their utility function increases.
In scenarios with a road accounting for two lanes the agents are placed in pairs, one in each lane. In this case the initial lateral distance of their pivot points is 3 meters, accounting for 1.2 meters of physical distance, agent-to-agent. In the scenarios with obstacles blocking lanes, this forces the agents to merge onto a populated lane. This can only be achieved though cooperative behavior (Bae et al., 2020). Thus the achievement of the objectives as outlined above gives a good indication of cooperativeness.

Results

The analysis is carried out two-fold: At first a sensitivity analysis is performed, followed by a pattern-based evaluation. The sensitivity analysis also serves the objective of finding those parameters where the FSX agents perform the best. These parameters can then be used for further analysis.
Following Grimm et al. (2005) the model is compared to the well adopted model of Gipps (1981) using patterns. In fact, Gipps' (1981) model is itself validated against patterns appearing in real-world traffic. Those patterns are what we aim to fit our model to as well. The pattern-based analysis builds the second part of the results section.

Sensitivity Analysis

The sensitivity analysis covers all three parameters introduced through the Future State Maximization algorithm as given in the methods section. The parameters listed in Table 1 are considered as constant, as the influence of these is beyond the scope of this report. However, it could be expected that a general reduction in maximum speed (i.e. a speed limit) reduces the possible risk taken by agents. Such an effect may be comparable to the increase of the decision sampling rate as shown below. Ultimately, the parameters taken into consideration are:

- N - The number of walkers
- τ - The horizon of future states
- α - Parameter balancing exploitation versus exploration

Additionally, the resolution of the temporal dimension has been doubled to investigate a higher sample rate of the decision making process. The result of every set of parameters has been calculated as the mean of a total of thirty simulations. Thus, we account for uncertainty introduced through the randomized nature of the Monte-Carlo like search algorithm of the walkers (Cerezo and Ballester, 2018; Hammersley, 2013).

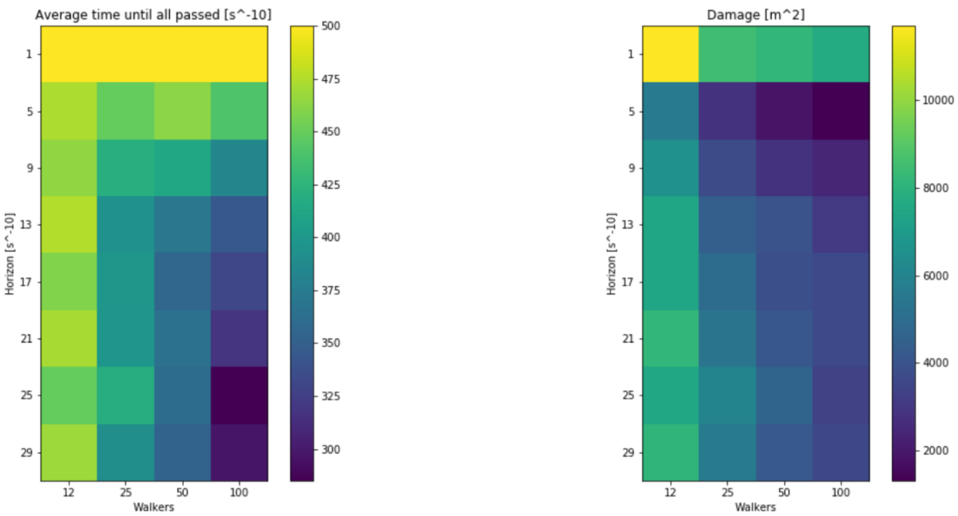

Figure 4: Average time for completion as well as damage caused without modifications to the agents. Both plotted over number of walkers N and horizon τ.

The scenario we chose to assess here, was the primary scenario as given in the scenarios section. Thus, all agents share the same parameters except for their starting positions. To measure the impact of the parameters on the success of the agents circumventing an obstacle, we collect two results:

- Time - The time it takes for the last agent to go beyond the $60m$ mark
- Damage - The total of intersecting area in between agents and non-accessible areas (see section Environment)

The damage value gives us a rough estimate of how aggressively the agents are driving. Thus, a lower damage value is preferred. The temporal aspect however, is the primary objective. Faster is better in this case. We are going to see that both values show some similarities in the location of their optima.

Horizon and number of walkers

As shown in Figure 4 when comparing the number of walkers N against the horizon τ, there exists a rather linear relationship. The quickest ways around the obstacle are found by using a maximum of walkers and horizon; Thus, scanning the future state space as densely as possible. The damage also decreases with a rising number of walkers. However, it appears to increase slightly when the horizon is extended. This could be due to the effective density of walkers at horizon τ decreasing as the number of states to be scanned rises exponentially.

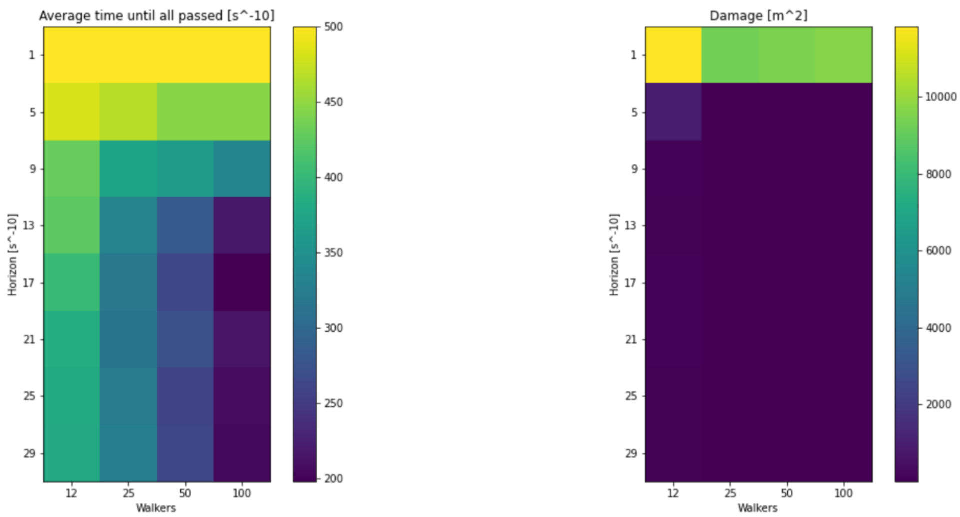

Figure 5: Average time for completion as well as damage caused by agents with dynamic horizon. Both plotted over number of walkers N and horizon τ.

This effect could be remedied by changing the algorithm slightly. By not only taking into account samples where walkers advanced up to τ the density of walkers could be optimized. This technique of a "dynamic slice" taken from the causal cone enables the agent to scan the state space more densely where it is necessary. The change in the result can be observed in Figure 5 and clearly shows the advance made herein.

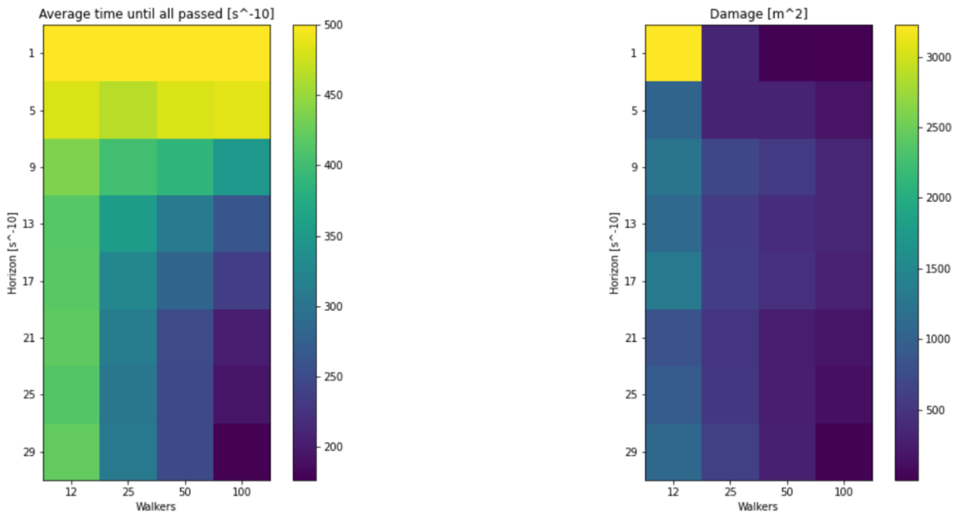

Figure 6: Average time for completion as well as damage caused by agents scanning their future state space twice as often. Both plotted over number of walkers N and horizon τ.

Prior to this finding, the one proposed solution was to increase the temporal resolution of the walker process. Thus, scanning not every second in the future state space but every half of a second. In effect, the scanning density as well as computational demand are increased. As expected, the results shown in Figure 6 improve. Especially at a low horizon, the agents profit from the additional temporal steps in their thought process.

Horizon and the alpha parameter

A non-linear effect could be observed in the first assessments of a graph showing the α value against the horizon τ. While a low horizon value yields inferior results as expected, the beneficial effect of a rising horizon is only significant in a certain regime. Moreover, the emergence of cooperation in between agents appears to rise and fall within the range of parameter α.

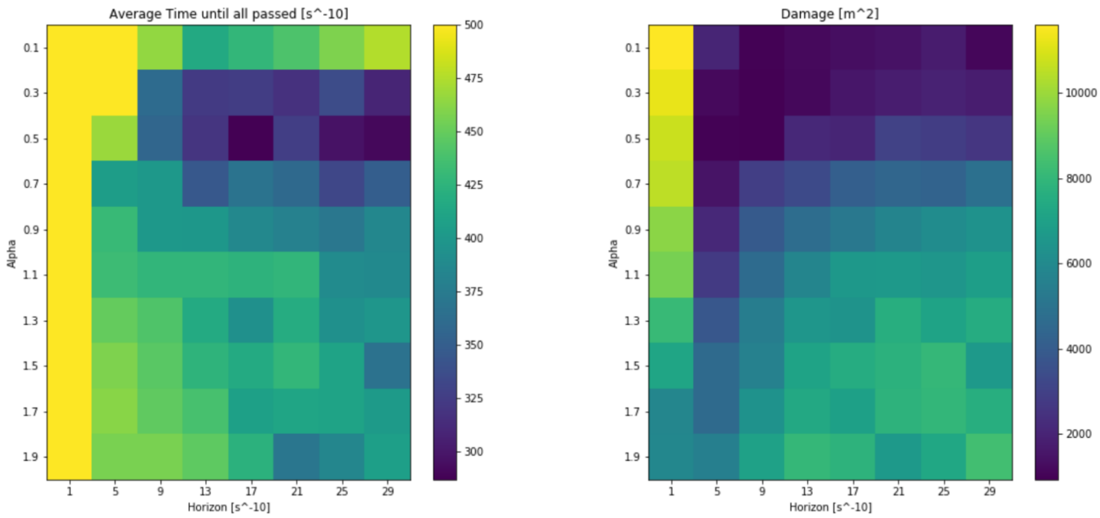

Figure 7: Average time for completion as well as damage caused without modification to the agents. Both plotted over α and horizon τ.

As visible in Figure 7 the range of ideal alpha values in our model appears to be in between 0.3 to 0.4. A very low α renders the agents unwilling to strive for their objective. On the other end, a higher α leads to detrimental effects of agents unwilling to cooperate with one another. Only by coordinating their respective advance through the small tunnel the agents can assure a good traffic flow and in effect low overall time and damage values.

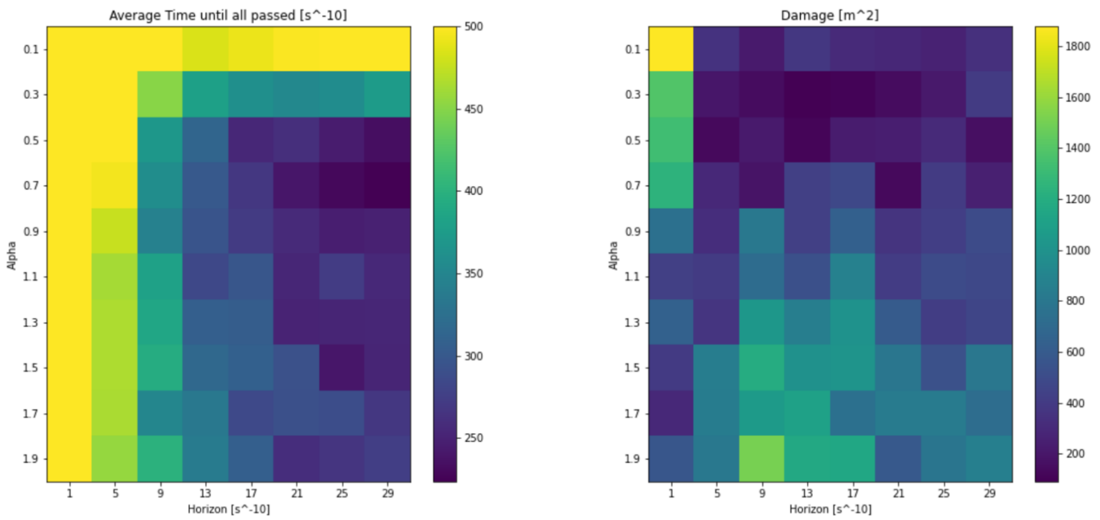

Figure 8: Average time for completion as well as damage caused by agents scanning their future state space twice as often. Both plotted over α and horizon τ.

Again doubling the amount of decision phases per time-step for the agents significantly improves the overall results. However, it does not lead to a qualitative change in the result as the non-linearity can still be spotted in Figure 8.

Patterns

Gipps (1981) presents two patterns featured by their car-following model: The horseshoe shaped speed-flow curve and the propagation of disturbances through traffic.

The horseshoe shape of the plotted flow given as vehicles per hour against average speed can also be observed in real-world traffic (Gipps, 1981). In fact, it appears intuitive that as average speed and vehicle density rise, the road capacity would induce a limit. With higher density then the speed has to decrease which in effect reduces the number of vehicles per hour.

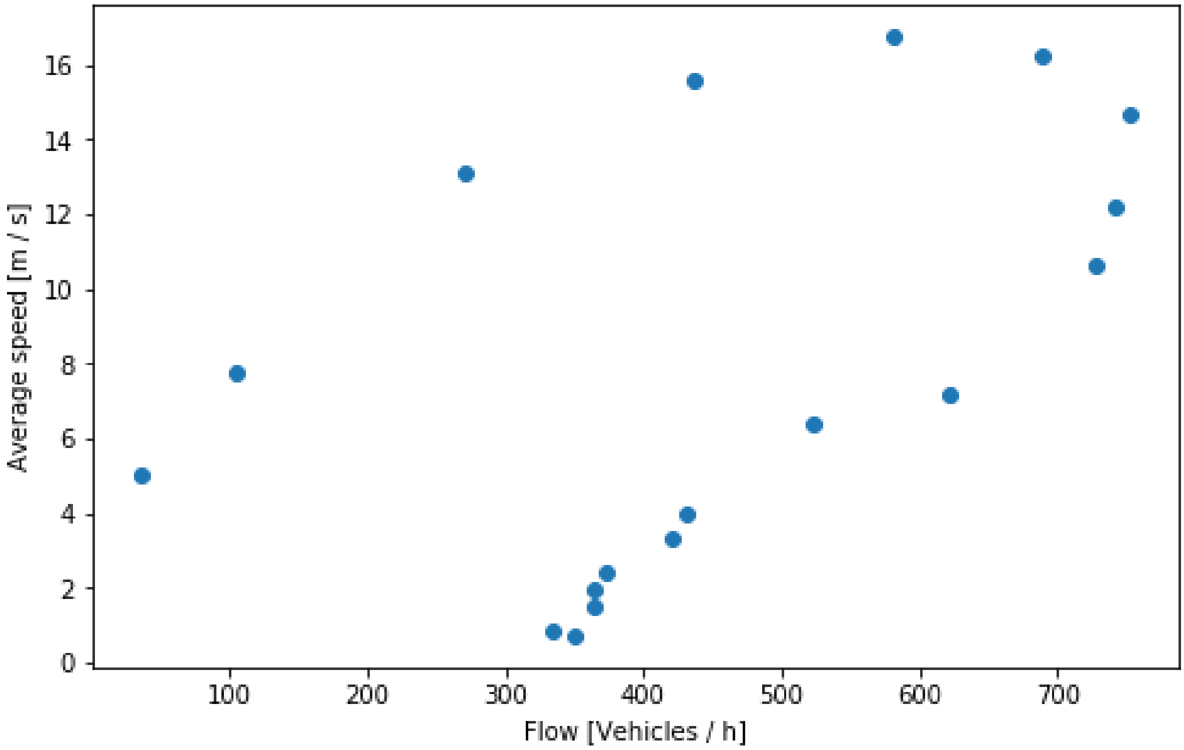

Figure 9: Horseshoe shaped Speed-Flow-Curve

For the assessment of this pattern, an open road is required. We therefore use the so called basic scenario as described in section "Scenarios" and given in Table 2. Using the parameters found to yield an optimal result at circumventing obstacles in our sensitivity analysis we can observe a similar pattern in our model shown in Figure 9. In comparison to Gipps' (1981) model, the absolute values differ slightly. However, they are within reasonable bounds and might only stem from the three lane road used by Gipps (1981) where our road only features two lanes of width.

To introduce disturbances into our model, we reduce the width of our road to only accommodate one car at a time. We then bring the foremost car to a halt and let it accelerate after a short stop and observe how the other agents react. The pattern observed in Figure 10 is also similar to the one shown in Gipps (1981), rendering the pattern oriented analysis successful.

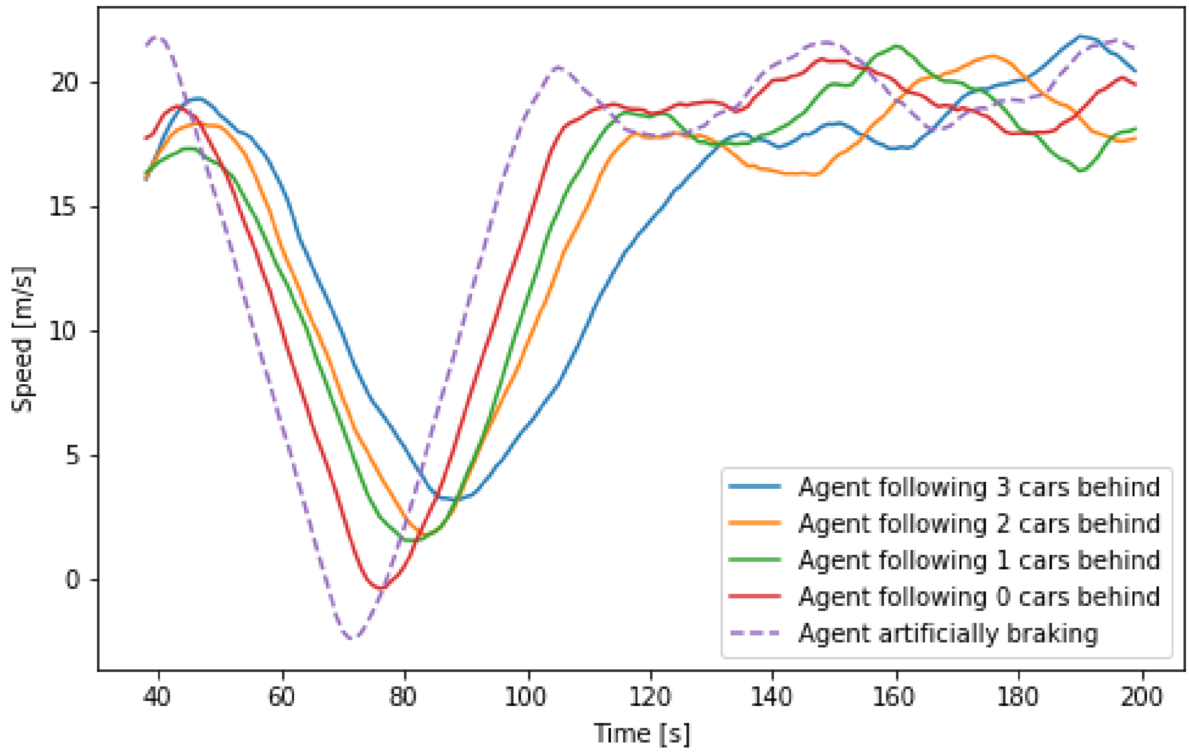

Figure 10: Reaction of the agents to an artificially caused perturbance in the traffic flow.

Discussion

Using an adaption of Cerezo and Ballester's (2018) algorithm, Future State Maximization reduces the amount of parameters to merely three factors: Horizon τ, number of walkers N and α. Additionally, with our adaption multiple objectives can be separated allowing for parameterization of single utility functions. While such undertakings are out of the scope of this work, the three parameters assessed in our sensitivity analysis show significant, while distinct influence in the agent's behavior.

Influence of the three FSX parameters

The horizon and number of walkers work together to increase the density of future state space scanning. Our analysis cannot show a significant difference in the influence of horizon versus the number of walkers. However, in more dynamic environments it would seem intuitive that a high number of walkers could not share the effect of a far horizon. As such, a better interpretation is that the horizon adjusts how much an agent anticipates the future, while the number of walkers is the actual adjustment of the scanning density.
This interpretation is especially true for our first algorithm, as it focuses on the final future state of a walker. Therefore, the states at the horizon outweigh previous future states. The effect is clearly visible in the sensitivity analysis as a higher horizon can introduce detrimental effects on the overall performance of the agents.
This effect is potentially removed by applying a "dynamic horizon" technique where the horizon is shifted to the last causal slice reached by living walkers smaller or equal to the initial horizon. This method could in effect also reduce the strong effect of a low number of walkers as it reduces the effect of unfortunate random walker decisions.
Unfortunately this dynamic horizon was a scanning strategy only found in a very late stage. Hence, it couldn't be assessed in sufficient detail to include a proper assessment of its effects in this report.

The α parameter shows a significant effect on the performance of the agents. In the case of a high alpha value the urgency of the agents to reach their goal on the very right of their map outweighs the strive for survival. In effect, the agents refuse to cooperate and often proceed to consecutively block one another. The upper bound of the alpha parameter in the sensitivity analysis was chosen to be 2. This value is in line with the interval given by Cerezo and Ballester (2018). However, it does not appear clear why a higher value should be disregarded. Nevertheless, it also seems unlikely that in our model any higher value of α would have lead to an increase of cooperation.

On the lower bound of α an interesting effect on cooperative behavior can be explored. The time the agents take to arrive at their goal increases as the α value decreases. The damage values also decrease only showing an increase for the lowest horizon considered.
However, while it is paramount to also assess the mistakes made by the agents, this measure could profit from improvements. If agents get locked into an unfavorable position the damage value can increase vastly while a fast agent might not collect much overlap with an opponent before it has crossed the area. As such a more realistic damage measurement incorporating impact velocity would be necessary for a closer look into the mistakes made by our agents.

Patterns

Future State Maximization appears to replicate the patterns found in Gipps (1981) successfully. It is, however, not clear if these patterns are not mere emergent phenomena found in any traffic system. So far no clear connections of the patterns pointing to specific human behavior has been found. Nevertheless, the patterns show some key features predominant in human driving. Especially the disturbance of the traffic flow shows clear signs of a reaction time. This is also implicit in the horseshoe pattern, as it slows down traffic flow in scenarios with high car density.

However, so far pattern-oriented modeling (Grimm et al., 2005) promises to give good qualitative results and it is only for a lack of patterns found to date that the model validation cannot be established further. Nevertheless, in Ciuffo et al. (2012) some patterns of real-world traffic are depicted providing input for future investigations into the qualitative aspects of our model.

Bounded-rational intelligence

The low number of parameters in conjunction with the intuitive definition of the degrees of freedom for our agents show how FSX could be a promising alternative to other artificially intelligent agents as in Jäger (2019) or rule-based ABMs as described by Epstein (1999). The damage values show that the agents are not perfectly rational but rather act in certain interests. Moreover, the agents appear to be acting independently as proposed in the definition of an agent by Grimm et al. (2005). So far only equivalent agents have been considered. The small number of parameters compared to e.g. an artificial neural network or an accumulation of several rules would make it trivial to define a heterogeneous population. In addition, the intuitive description of future state expectations also enables creating agents with specific beliefs. However, the latter could require considerate amounts of additional work in some cases.

Cooperation of FSX agents

A novel finding we can report is, that FSX agents can in fact stop cooperating. This could be closely related to the breakdown and respective emergence of structure within the collectivism of CEF agents as shown in Hornischer et al. (2019). What is novel in our investigations is the introduction of a utility function as pioneered by Cerezo and Ballester (2018). The separation of objectives we have conducted within this work enables us to independently steer the influence of both survival (exploration) and utility (exploitation). We argue that the alpha parameter alone for a single reward function encoding both survival and utility as previously stated by Cerezo and Ballester (2018) cannot configure this balance. Reducing alpha in this case also renders the agent incapable of striving for survival as all information about survival of the walkers gets lost.

With our modification however it has become evident, that for cooperation a balance has to be struck. Too high urgency to reach an objective yields agents too egoistic to make way for a fellow. With no agent acting cooperative, merging lanes becomes difficult (Bae et al., 2020) which leads to a collective delay. On the other end of the spectrum, low urgency leaves the agents indifferent enough about the objective to get stuck in sub-optimal situations. It seems that a small amount of urgency is required to collectively strive for the greater good.

Another reason why other researchers such as Charlesworth and Turner (2019) discovered collaboration of their agents could be due to the sequential evaluation of their agents’ decisions and high sample rates. As can be seen in Figure 6 and Figure 8 an increased number of decision steps improves the result. It effectively, reduces the reaction time to the actions of other agents. Due to the sequential calculation every agent has complete information about its surroundings. If the difference in between the time-steps is made small enough, the agents can effectively react to the smallest of dynamics and appear to be cooperative.

Nevertheless, a more purpose made model would be required to go deeper into cooperation of FSX agents. A microscopic traffic model as presented within this report can show the breakdown of this formerly discovered emergence but has only limited capacity of explaining it. Nevertheless, such a model could be applied in future investigations on the reaction of autonomous driving algorithms to bounded-rational driving behavior.

Computational demand

The computational demand of our method cannot compete with others. Its computational demand could render more elaborated investigations time consuming and expensive. It is thus only applicable in cases of few agent systems, or when reducing the option and/or state space to a low number of discrete members.

However, the research on this novel method is still in its infancy. It is likely that future research will decrease the computational demand. The parallel nature of the walker entities enables parallel and concurrent computing. One such frontier could be found in our "dynamic horizon" which could reduce the computational demand in certain situations. It is certainly a research topic worth investigating in the future as it could improve both computational and intelligent performance.

Moreover, the reuse of walker paths especially in discrete choice scenarios seems worthwhile of investigation. A plethora of research questions arises from this and other current shortcomings in FSX-ABMs.

Conclusions

The Future State Maximization principle shows qualities making it applicable to agent-based modeling. The agents exhibit what could be interpreted as intelligent behavior. Moreover, this behavior can be adjusted by varying the horizon and the number of walkers an agent spawns. Furthermore, utilizing the algorithm of Cerezo and Ballester (2018) a utility function can be introduced. An additional parameter α is available to adjust the influence of this function. We successfully extended this algorithm to separate survival from utility dynamics.
As the sensitivity analysis shows, these parameters are rather influential. Therefore, they are useful for fitting the model to data.

An assessment of microscopic traffic modeling of small road sections using FSX agents shows qualitative similarities to the common car-following model of Gipps (1981). Furthermore, patterns observed in real-world traffic can also be observed when using FSX agents. However, in contrast to Gipps' (1981) model, the behavior of these agents is emergent from their thought process using the Future State Maximization principle and the (physical) boundaries applying to them.

Bounded rationality is implicit in our model. We can report that the FSX principle is suitable for agent-based modeling in this regard, as the cooperation can in fact break down due to limited rationale of the agents.
Therefore, the agents can exhibit behavior which can be utilized for models of lacking traffic regulations such as shared spaces. Moreover, scenarios where some agents do not totally abide to the law are plausible by utilizing Future State Maximization for their decision dynamics.

With our sensitivity analysis we could establish some foundation for the understanding of the social behavior within future state maximizing agents. However, for a more clear insight into the interaction of individual FSX agents a more purpose made model would be beneficial. Future research could employ simpler models such as a public-good game to strengthen the insights into cooperation and its breakdown among such agents.

Index of Tables

Table 1: Parameters for the car-agent properties.....5

Table 2: List of scenarios and the properties of their environments.....11

Table of Figures

Figure 1: The causal cone of an agent and walkers scanning it up to the agent's horizon τ. The agent’s state is depicted as red dot at the tip of the cone, whereas the walkers are shown as green strings. At t_s a causal slice can be observed.....3

Figure 2: Two agents and their velocity before and after they have come to a decision. The orange agent decides to turn left, while the green agent decides to decelerate.....6

Figure 3: Five agents approaching an obstacle blocking the right lane in the primary scenario.....8

Figure 4: Average time for completion as well as damage caused without modifications to the agents. Both plotted over number of walkers N and horizon τ.....12

Figure 5: Average time for completion as well as damage caused by agents with dynamic horizon. Both plotted over number of walkers N and horizon τ.....13

Figure 6: Average time for completion as well as damage caused by agents scanning their future state space twice as often. Both plotted over number of walkers N and horizon τ.....14

Figure 7: Average time for completion as well as damage caused without modification to the agents. Both plotted over α and horizon τ.....15

Figure 8: Average time for completion as well as damage caused by agents scanning their future state space twice as often. Both plotted over α and horizon τ.....15

Figure 9: Horseshoe shaped Speed-Flow-Curve.....16

Figure 10: Reaction of the agents to an artificially caused perturbance in the traffic flow.....17

References

Bae, S., Saxena, D., Nakhaei, A., Choi, C., Fujimura, K., Moura, S., 2020. Cooperation-Aware Lane Change Maneuver in Dense Traffic based on Model Predictive Control with Recurrent Neural Network, in: 2020 American Control Conference (ACC). Presented at the 2020 American Control Conference (ACC), pp. 1209–1216. https://doi.org/10.23919/ACC45564.2020.9147837

Bilbao, I., Bilbao, J., 2017. Overfitting problem and the over-training in the era of data: Particularly for Artificial Neural Networks, in: 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS). Presented at the 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 173–177. https://doi.org/10.1109/INTELCIS.2017.8260032

Bonabeau, E., 2002. Agent-based modeling: Methods and techniques for simulating human systems. Proc. Natl. Acad. Sci. 99, 7280–7287.

Bora, Ş., Evren, V., Emek, S., Çakırlar, I., 2019. Agent-based modeling and simulation of blood vessels in the cardiovascular system. SIMULATION 95, 297–312. https://doi.org/10.1177/0037549717712602

Bousso, R., Harnik, R., Kribs, G.D., Perez, G., 2007. Predicting the cosmological constant from the causal entropic principle. Phys. Rev. D 76, 043513. https://doi.org/10.1103/PhysRevD.76.043513

Cazzolla Gatti, R., Koppl, R., Fath, B.D., Kauffman, S., Hordijk, W., Ulanowicz, R.E., 2020. On the emergence of ecological and economic niches. J. Bioeconomics 22, 99–127. https://doi.org/10.1007/s10818-020-09295-4

Cerezo, S.H., Ballester, G.D., 2018. Fractal AI: A fragile theory of intelligence. CoRR abs/1803.05049.

Charlesworth, H.J., Turner, M.S., 2019. Intrinsically motivated collective motion. Proc. Natl. Acad. Sci. 116, 15362. https://doi.org/10.1073/pnas.1822069116

Ciuffo, B., Punzo, V., Montanino, M., 2012. Thirty Years of Gipps' Car-Following Model. Transp. Res. Rec. J. Transp. Res. Board 2315, 89–99. https://doi.org/10.3141/2315-10

Ebmeier, F., 2017. Simulations of pedestrian flow and lane formation through causal entropy. Georg-August-Universität Göttingen.

Epstein, J.M., 1999. Agent-based computational models and generative social science. Complexity 4, 41–60. https://doi.org/10.1002/(SICI)1099-0526(199905/06)4:5<41::AID-CPLX9>3.0.CO;2-F

Gipps, P.G., 1981. A behavioural car-following model for computer simulation. Transp. Res. Part B Methodol. 15, 105–111. https://doi.org/10.1016/0191-2615(81)90037-0

Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W.M., Railsback, S.F., Thulke, H.-H., Weiner, J., Wiegand, T., DeAngelis, D.L., 2005. Pattern-Oriented Modeling of Agent-Based Complex Systems: Lessons from Ecology. Science 310, 987–991. https://doi.org/10.1126/science.1116681

Hammersley, J., 2013. Monte Carlo Methods. Springer Science & Business Media.

Hidas, P., 2002. Modeling lane changing and merging in microscopic traffic simulation. Transp. Res. Part C Emerg. Technol. 10, 351–371. https://doi.org/10.1016/S0968-090X(02)00026-8

Hornischer, H., Herminghaus, S., Mazza, M.G., 2019. Structural transition in the collective behavior of cognitive agents. Sci. Rep. 9, 12477. https://doi.org/10.1038/s41598-019-48638-8

Ishii, S., Yoshida, W., Yoshimoto, J., 2002. Control of exploitation–exploration meta-parameter in reinforcement learning. Neural Netw. 15, 665–687. https://doi.org/10.1016/S0893-6080(02)00056-4

Jäger, G., 2019. Replacing Rules by Neural Networks A Framework for Agent-Based Modelling. Big Data Cogn. Comput. 3, 51. https://doi.org/10.3390/bdcc3040051

Klyubin, E.S., Polani, D., Nehaniv, C.L., 2005. Empowerment: A universal agent-centric measure of control, in: In Proc. CEC. IEEE.

Kosko, B., 1986. Fuzzy cognitive maps. Int. J. Man-Mach. Stud. 24, 65–75. https://doi.org/10.1016/S0020-7373(86)80040-2

Kotusevski, G., Hawick, K.A., 2009. A review of traffic simulation software.

Manson, S., An, L., Clarke, K.C., Heppenstall, A., Koch, J., Krzyzanowski, B., Morgan, F., O'Sullivan, D., Runck, B.C., Shook, E., Tesfatsion, L., 2020. Methodological Issues of Spatial Agent-Based Models. J. Artif. Soc. Soc. Simul. 23, 3.

Martyushev, L.M., Seleznev, V.D., 2006. Maximum entropy production principle in physics, chemistry and biology. Phys. Rep. 426, 1–45. https://doi.org/10.1016/j.physrep.2005.12.001

Melis, A.P., Hare, B., Tomasello, M., 2006. Engineering cooperation in chimpanzees: tolerance constraints on cooperation. Anim. Behav. 72, 275–286. https://doi.org/10.1016/j.anbehav.2005.09.018

Rockett, R.J., Arnott, A., Lam, C., Sadsad, R., Timms, V., Gray, K.-A., Eden, J.-S., Chang, S., Gall, M., Draper, J., Sim, E.M., Bachmann, N.L., Carter, I., Basile, K., Byun, R., O'Sullivan, M.V., Chen, S.C.-A., Maddocks, S., Sorrell, T.C., Dwyer, D.E., Holmes, E.C., Kok, J., Prokopenko, M., Sintchenko, V., 2020. Revealing COVID-19 transmission in Australia by SARS-CoV-2 genome sequencing and agent-based modeling. Nat. Med. 26, 1398–1404. https://doi.org/10.1038/s41591-020-1000-7

Tang, T.-Q., Huang, H.-J., Shang, H.-Y., 2015. Influences of the driver's bounded rationality on micro driving behavior, fuel consumption and emissions. Transp. Res. Part Transp. Environ. 41, 423–432. https://doi.org/10.1016/j.trd.2015.10.016

Van Dyke Parunak, H., Savit, R., Riolo, R.L., 1998. Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide, in: Sichman, J.S., Conte, R., Gilbert, N. (Eds.), Multi-Agent Systems and Agent-Based Simulation, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 10–25. https://doi.org/10.1007/10692956_2

Wissner-Gross, A.D., Freer, C.E., 2013. Causal Entropic Forces. Phys. Rev. Lett. 110, 168702. https://doi.org/10.1103/PhysRevLett.110.168702